REMARKS

Claims 1-5, 7, 11-14 and 20-30 are pending in the present application. No claims are being amended or added via this response.

Applicants would like to thank the Examiner for the interview held on October 9, 2003 to discuss distinctions between the pending claims and cited prior art. The following discussion of patentablility is consistent with the interview held on October 9, 2003. Allowance of the pending claims is respectfully requested.

The following remarks address the rejections of claims 1-5, 7, 11-14 and 20-30 as set out by Examiner in the Final Office Action mailed on August 6, 2003.

Rejections of Claims 1-4, 11-13 and 20-30 under 35 U.S.C. § 102

The Examiner has rejected claims 1-4, 11-13, and 20-30 under 35 U.S.C. § 102(e) based on the teachings of Saylor, et al., (U.S. Patent 6,501,832).

Applicants would like to point out the distinctions between claimed invention and the Saylor reference as discussed in the interview of October 9, 2003. Claim 1 recites that the compiler operates to compile a document (as retrieved by a fetcher) into compiled document data in executable form. The compiled document data is stored in a cache prior to receiving a request by a user for the text-based document. Thus, each time a user requests a text-based document, the interactive voice response system need not recompile the requested text based document to service the request.

Saylor '832 discloses a technique of enabling users to access information using a voice code (Vcode), which is a code assigned to a particular raw page of

content that is delivered to a user via a phone (column 1, lines 58-61). The Vcode may be an address or identifier of a corresponding Vpage (i.e., a raw XML web page) stored in memory. Based on receipt of a Vcode associated with requested content, a Vpage (a raw web page) is retrieved from database 18 or retrieved over a network 20 from Vpage servers 22 (column 18, lines 23-27) depending on whether it is locally available. An important point in Saylor is that a Vpage is a raw web page such as that based on VoiceXML. Each time a Vpage is retrieved, the raw data of the Vpage must be compiled (e.g., interpreted) before servicing a user's request.

Vpages (web pages in Voice XML format) are stored in database 18 where they are retrieved as shown in FIG. 2 of Saylor. During operation, Vpage retrieval system 32 retrieves unprocessed Vpages (web pages) that are in turn processed in response to the Vcode by Vpage menu module 36 and Vpage execution module 34. This is discussed in detail in corresponding text at column 18, lines 45-65. Upon retrieval of the Vpage corresponding to a given Vcode, Vpage execution module 34 interprets (e.g., compiles) retrieved Vpage (webpage) content provided by Vpage retrieval system 32. As discussed in Saylor '832, execution (i.e., XML interpretation) of the web page content includes scanning the Vpage for certain (XML) tags and generating menus depending on the Vpage which is generally a compiling or interpreting process. This conventional technique is generally described in the background of the present application referencing Figure 1. In Saylor '832, the web pages or Vpages in unprocessed XML format are optionally stored and retrieved from a local data base 18 or a remote server.

In a nutshell, voice network access provider (VNAP) 12 and, more specifically, Vpage (web page) menu module 36 in FIG. 2 of Saylor utilizes the collection of Vpages (or web pages in XML format) stored in database 18 to generate an appropriate audible response in real-time to user 14. Note that

Vpage menu module 36 includes a VoiceXML interpretation module to produce the appropriate audible response (column 18, lines 59-65). Accordingly, in Saylor, each retrieved Vpage is compiled in real-time by the Vpage module 36 and Vpage execution module 34 in response to a Vcode received from a user during a phone call.

In contradistinction to this cited technique of re-compiling a raw Vpage each time it is retrieved as in Saylor '832, claim 1 of the present application recites that a compiler converts documents (such as web page documents originally in XML format) into compiled document data (such as executable code) that is stored in a cache. The fetcher retrieves the appropriate compiled document data from the cache to satisfy a user request. More specifically, an execution thread services the user request by executing the compiled document data stored in and retrieved from the cache. Based on this technique of storing pre-processed data documents (such as web pages written in a markup language), a proper response such as an audio reply can be more quickly generated for a particular user because the executable file associated with a requested document (such as an VoiceXML page) need only be retrieved from cache and executed by execution thread to generate the response. Thus, an execution thread can skip a step of processing or interpreting raw document data such as a Voice XML web page before responding to an incoming request. As discussed, according to Saylor '832, a Vpage must be 'interpreted' or 'compiled' to generate an executable file (such as an audio file) each time a user requests a particular document (e.g., during a phone call).

The Examiner has likened the 'compiler' in claim 1 to content interpreter 66 (FIG. 3) as discussed in Saylor at column 21 lines 20-22. It is respectfully submitted that although Vpage server 22 does include interpreter 66, there is no mention, teaching or suggestion that 'interpreted' or 'compiled' documents (or Vpages) generated by interpreter 66 are stored or cached in memory such as

database 18. Instead, the raw uninterpreted Vpages (XML web page documents) themselves are stored in database 18. As its name suggests and according to the associated text in Saylor, Vpage (webpage) retrieval system 32 (FIG. 2) only retrieves Vpages (raw webpages) from database 18 or web page server 22 that thereafter must be processed. Saylor does not teach, disclose or suggest storing already interpreted document data as executable code as in the claimed invention. For example, already interpreted or compiled documents in executable form is stored in a cache as recited in claim 1 so that an execution thread need only execute the code in response receiving to a call from a user.

On a final note, the Office Action also cites column 4 lines 16-28 to reject the claimed invention. Applicants would like to point out that Saylor discloses merely storing content in various formats as opposed to the actual compiled raw Vpages. Consistent with the above discussion, the invention as in claim 1 is distinguished because it involves storing compiled documents prior to receiving a request for corresponding audio information.

It is respectfully submitted that in view of the above amendment and remarks, claim 1 is novel and non-obvious as it incorporates techniques contrary to previously accepted wisdom and blueprints for the inventive method cannot be found in the individual or combined cited references. Accordingly, it is submitted that independent claim 1 and corresponding dependent claims 2-4 are in condition for allowance over the prior art.

Claim 11 includes similar limitations as discussed above for claim 1. For applicable reasons, it is submitted that independent claim 11 and corresponding dependent claims 12-14 are in condition for allowance.

Claim 3 recites that the voice response system of the present invention includes a backup interpreter that otherwise provides a response to a user in the

event of a failure. The Examiner has likened the 'backup interpreter' (e.g., a secondary interpreter in case a primary interpreter experiences a failure) in claim 3 to Vpage (webpage) menu module 36 (FIG. 2) as discussed in Saylor at column 18 lines 56-65. It is respectfully submitted that this cited passage does not discuss the claimed technique of (nor does it appreciate the technical hurdles associated with) providing a backup or secondary interpreter to provide a response in the event of a failure. In fact, Saylor does not even address the issue of how to provide continued service in the event of a failure. Additionally, Saylor mentions use of a single interpreter and therefore does not discuss use of a backup or secondary interpreter. The claimed invention therefore includes a limitation not found in the cited prior art. This technique of providing backup in the claimed invention increases overall reliability of the voice response system in the event of failures. Allowance of claim 3 is also respectfully requested.

Rejections of Claims 5 and 7 under 35 U.S.C. § 103

The Examiner has rejected claims 5 and 7 under 35 U.S.C. § 103(a) based on the teachings of Saylor, et al., (U.S. Patent 6,501,832) in view of Paleiov, et al, (U.S. Patent 6,560,320).

It is respectfully submitted that claim 5 includes distinguishing limitations over Saylor and Paleiov. For example, claim 5 recites that the fetcher of the voice response system is operative to retrieve compiled documents (instead of raw XML web pages) similar to the limitation discussed in claim 1. Saylor '832 discusses retrieving raw web pages that must be compiled prior to providing an audio response to a user, not compiled documents as in the claimed invention. Fetching compiled documents instead of raw XML pages reduces the need for redundant pre-processing of raw web pages each time they are retrieved, thus reducing an amount of time to respond to a user request.

Additionally, the interactive voice response system of claim 5 recites a backup interpreter (e.g., a secondary interpreter operating as a backup to a first interpreter) that otherwise provides a response to a user in the event of a failure. Based on use of the backup Voice XML interpreter and state information of executed compiled document stored in the storage device, the interactive voice response system of the present invention can provide continuous service even in the event of a failure such as a primary interpreter failure. For example, the state information may identify status information (i.e., state information) such as a point at which a program is being executed to respond to a user of the voice response system. In the event of a failure, the backup interpreter (when switchover occurs) in claim 5 may identify, e.g., based on state information, a proper point of program execution to provide continued service without a user ever being aware of the switchover to the backup interpreter. Thus, the backup interpreter may pick up when a primary interpreter experiences a failure.

The Examiner has likened the 'backup interpreter' (secondary interpreter) in claim 5 to the XML-based Voice content interpreter 66 (FIG. 3) as discussed in Saylor at column 21 lines 20-29. It is respectfully submitted that this cited passage does not discuss the claimed technique of (nor does it appreciate the technical hurdles associated with) providing a backup or secondary interpreter to provide a response in the event of a failure. Instead, the cited passage recites use of only a single interpreter to process raw XML web pages (e.g., Vpages). Therefore, the claimed invention includes a limitation not found in the cited prior art.

The Examiner agrees that Saylor fails to disclose executing a document in the event of a failure. For this limitation, the Examiner cites Paleiov column 5, lines 62-65. The cited passage in Paleiov merely points out that in the event of a failure in a protocol handshaking, "then IVR responds to the user with a sequence of voice prompts." Note that the previous paragraph (column 5, lines

45-61 of Paleiov) describing use of a graphic display. The cited paragraph (column 5, lines 62-65) merely states (after stating "on the other hand") that, in the alternative to paragraph in column 5 lines 45-60, "IVR may respond with voice prompts as known in the art" for audio-only connections. Thus, Paleoiv neither teaches nor suggests an interactive voice response system including a backup interpreter as discussed above which is capable of providing seamless switchover in the event of a failure on a primary interpreter.

The technique of providing backup processing in claim 5 increases overall reliability (e.g., continuous support to a user) of the voice response system in the event of failures is not taught or suggested by the individually or combined cited references. Allowance of claim 5 and corresponding dependent claim 7 is therefore also respectfully requested.

Patentability of Claims 20-30

Claim 20 includes the limitation that a cache stores executable code (e.g., compiled document data) associated with text-based documents and the fetcher searches and retrieves a cache for executable code to satisfy user requests. As discussed above with respect to claim 1, voice network access provider 12 in FIG. 2 of Saylor '832 does not store executable code associated with a webpage (e.g., XML based) document in stored memory. For similar and applicable reasons as discussed above, claim 20 is distinguished and advantageous over the cited references. Allowance of claim 20 and corresponding dependent claims 21-29 is respectfully requested.

Consistent with the interview on October 9, 2003, Applicants would like to point out that claim 21 further distinguishes the claimed invention (of claim 20) over the prior art because it recites that a compiler converts the text-based document (such as XML web page information) into executable speech code for storage in the cache prior to receipt of the incoming request. Based on this

technique in claim 21, compiled or executable code associated with a previously requested text-based document is stored in cache for later retrieval. Thus, a text-based document such as a web page need not be completely recompiled at run rime when another user requests the same text-based document. Rather, the previously compiled executable code is retrieved from cache and executed to service each successive request.

Claim 22 further distinguishes the claimed invention (of claim 20) over the prior art because it recites that the fetcher initiates a communication with a remote server in order to retrieve a text-based document such as a web page if a corresponding executable code is not stored in the cache. As discussed, Saylor does not search a cache for executable code to service a user. Nor does Saylor initiate communication with a remote server if executable code is not stored in cache. In contradistinction to Saylor, the fetcher in claim 22 initiates communication with a remote server to retrieve a document such as raw web page content. This conditional procedure not recited in Saylor increases flexibility because the claimed voice response system services a request by retrieving an executable file from cache or retrieving raw document data from a remote server. Thus, the technique as recited in claim 22 is distinct and advantageous over the cited prior art.

Claim 23 further distinguishes the claimed invention (as in claim 22) over the prior art because it recites that a compiler converts retrieved text-based documents (e.g., XML pages) into corresponding executable code that is stored in cache. Based on this technique of pre-compiling a retrieved text-based document, fetcher need only retrieve and execute the executable code to service another user requesting a same text-based document as already processed by the compiler. Saylor does not address this technical hurdle anywhere in his issued patent. Thus, it is respectfully submitted that claim 23 is neither

anticipated nor obvious in light of the prior art. Consideration and allowance of new dependent claim 23 is also respectfully requested.

Claim 24 further distinguishes the claimed invention (as in claim 20) over the prior art because it recites that executable code (or compiled documents) in cache is utilized by multiple execution threads to provide a response to multiple users. Based on this technique of compiling a retrieved text-based document, fetcher need only retrieve and execute the executable code in cache in the event that multiple users simultaneously request a particular text-based document. Saylor also does not address this technical hurdle of simultaneously servicing multiple users using a common executable code in cache. Thus, it is respectfully submitted that claim 23 is neither anticipated nor obvious in light of the prior art. Consideration and allowance of new dependent claim 23 is also respectfully requested.

Conclusion

In view of the foregoing remarks and interview held October 9, 2003, it is respectfully submitted that the claims of the present application are in condition for allowance. A Notice to this affect is respectfully requested. If the Examiner believes, after submission of this reply, that the Application is not in condition for allowance, the Examiner is respectfully requested to call the Applicants' Representative at the number below.

Applicants hereby petition for any extension of time which is required to maintain the pendency of this case. If there is a fee occasioned by this response, including an extension fee, that is not covered by an enclosed check, please charge any deficiency to Deposit Account No. 50-0901.

If the enclosed papers or fees are considered incomplete, the Patent Office is respectfully requested to contact the undersigned Attorney at (508) 366-9600, in Westborough, Massachusetts

Respectfully submitted,

Paul P. Kriz, Esq.
Attorney for Applicants
Registration No.: 45,752
CHAPIN & HUANG, L.L.C.
Westborough Office Park
1700 West Park Drive
Westborough, Massachusetts 01581
Telephone: (508) 366-9600
Facsimile: (508) 616-9805

Attorney Docket No.: NMS03-05

Dated: October 15, 2003